

The Role of Reactive and Game Tasks in Competitions

Ilia NINKA

*Department of Computer Science, Tirana University
Zog i Parë blvd., Tirana, Albania
e-mail: ilia.ninka@yahoo.com*

Abstract. The need for more attractive tasks in teaching algorithms and in informatics competitions attracts many authors to reactive tasks as a powerful tool that created in student a desire for coping with hard tasks. In comparison with batch tasks, reactive tasks and especially the programming of games are very challenging, very comprehensive and fit perfectly with the story. In this paper an attempt to point out some attributes and priorities of reactive tasks in general, and game tasks in particular, versus batch ones is made.

Key words: batch task, reactive task, game programming, story, input data.

Teaching programming and algorithms in higher education is related with many difficulties both for the teacher and the students. The students hardly understand the need for complicated constructions and data structures related to the programming language, seeing that most of the problems presented by the teacher are elementary or may be solved by easier tools; spreadsheets, for example. The teacher is also faced with the difficulty for providing the students real world data for convincing them that it is necessary to write a program to develop the data and find the solution.

“Programming games will encourage students to learn more, and to apply what they learn to create new things, reaching the ultimate goal of education. Through the establishment of programming games as a core curriculum of Computer Science classes, students will learn algorithms faster and with a deeper understanding, and will want to do this because of the fun and accomplishment associated with the creation of a computer game” (Baibak and Agrawal, 2007).

During the process of creating algorithmic nature tasks informatics competitions there are some restrictions to be respected. We will consider them one by one from the view point of the topic of this paper.

According to the IOI 2008 competition rules, the tasks could be of the following types:

- **Batch tasks:** Solutions comprise a single source file of a computer program which reads data from the standard input (stdin) and writes its answer to the standard output (stdout).
- **Reactive tasks:** Solutions comprise a single source file of a computer program that is compiled together with an “opponent” library provided by the organizers, and

interacts with it according to the specification given in the task description. Such solutions are not allowed to read anything from the standard input, or write anything to the standard output.

- **Output-only tasks:** Solutions comprise a set of “output” data files. The contestants submit a zip or tgz archive file containing some or all the output data files.

In this paper game tasks and reactive tasks are considered from a common point of view. Output-only tasks are not considered.

1. The Story

Each task in the IOI, traditionally, is described as a real life situation and in some cases with characters endowed with real names and real habits. Sometimes, due to the diversity of participants, these create undesired ethic or moral situations.

Due to this tradition it happens that after having developed a task the author invents a story to fit it as much as possible. Sometimes the invented story is successful but there are cases that the story does not fit the problem as expected.

The students at first have to throw off the story and to discover the real problem to be solved and programmed. Sometimes this is quite natural, when the story and the problem are in harmony, but it happens that this may not be so easy especially when the story does not fit well with the problem.

When the task is a game one there is no need for a story. The story and the problem are the same. For the student this is a quite clear situation; he loses no time in discovering the problem behind the story, but only has to think how to solve the situation. In such case the student feels motivated because this situation is similar with other game situations which they had to surpass since childhood and they have some prior experience in such situations even without to the necessity of using computers or programming.

2. The Size of Input Data

In almost all competitions the input data is a real problem in itself when very large files are to be constructed. The concern for such large files is related to the aim to estimate the efficiency of algorithm used by the student. Sometimes the enormous quantity of input data is far from being a natural description of a real world situation as the story pretends to give. In reactive tasks, especially in game tasks, the input data is not such a concern for the author.

Let see some of game tasks given in the IOI. In these tasks the input data are quite natural and fit perfectly with the story.

a) **Task 4** (Long-list of tasks, IOI'1990, Minsk, Belarus). *Given integer number K . A strip of paper is divided into N cells ($K \leq N \leq 40$). Two players choose and cross out K empty adjacent cells one by one. The winner is the one who has made the last move.*

In this task the input is **only two integers** K, N , where $K \leq N \leq 40$.

b) Task **RUBIK'S TOOLKIT** (IOI' 1992, Bonn, Germany). Write a program that allows the user to repeatedly solve any of the given three sub-problems ... in any order. You may assume that the length of each input string is at most 35.

We escaped formulation of the three sub-problems which is long enough, but in this task the input data are quite reasonable, only 35 characters!

c) Task **LETTER GAME** (IOI' 1995, Eindhoven, the Netherlands) . . . **Input Data.** The input file *INPUT.TXT* contains one line with a string of lowercase letters (from 'a' to 'z'): the letters collected. The string consists of at least 3 and at most 7 letters in arbitrary order. The "dictionary" file *WORDS.TXT* consists of at most 40,000 lines. At the end of this file is a line with a single period ('.'). Each of the other lines contains a string of at least 3 and at most 7 lowercase letters. The file *WORDS.TXT* is sorted alphabetically and contains no duplicates.

In this task the input data to be faced, while relatively large, is a common dictionary that the students use in their daily work in school.

d) Task **A GAME** (IOI' 1996, Veszprém, Hungary) ... **Input Data.** The first line of file *INPUT.TXT* contains the size N of the initial board. N is even and $2 \leq N \leq 100$. The remaining N lines contain one number in each line, the contents of the initial board in left to right order. Each number is at most 200.

In this task a common game board is supposed to have no more than 200 numbers.

e) Task **MAGIC SQUARES** (IOI' 1996, Veszprém, Hungary) ... **Input Data.** The file *INPUT.TXT* contains 8 positive integers in the first line, the description of the target configuration.

f) Task **THE GAME OF HEX** (IOI' 1997, Cape Town, South Africa) Your program must not read from or write to any files. Your program must not receive keyboard input, and must not produce output on the screen. It will receive all its input from the functions in the hex library.

As it is seen from these tasks the input data are a complement of the task itself. This may not be the case in some batch tasks. For the story's sake the authors sometimes go so far that reality is forgotten making the story sound very strange! Let us consider only one batch task:

g) Task **SEEING THE BOUNDARY** (IOI' 2003, Kenosha, USA).

Now let us examine farmer's Don *field*. It is $500 \text{ km} \times 500 \text{ km} = 250\,000 \text{ km square!}$ This is almost the surface of Italy! But what about the rocks! This looks not as a farm but as a stone depository with as many as 30 000 huge rocks! No machine could do any agricultural work in this field! But what about the farmer Don himself: he is frightened by the fact that he must be cautious not to touch the rock, not to stand *within* a rock, and not to stand on a rock!

3. Inventing Strategy

Being quite natural and endowed with a rich flavor of challenge, the reactive tasks and game tasks arouse the interest of the student for not only trying to win a game but to

discover the best algorithm that ensures the victory when they have to move first. Even when the contest is finished, students are more biased to discuss game tasks with the aim to discover what they missed doing during the contest. This was for example a case when in the transit area in the airport; while waiting for the flight the students continued the game and discovered a quite simple algorithm.

Nearly all games require seeing patterns, making plans, searching combinations, judging alternative moves, and learning from experience – all being skills which are also involved in many daily tasks. So, Ginsberg (1998) was right when declared: “More than just competing with people, game-playing machines complement human thinking by offering alternative methods to solving problems”.

4. Not Only Competition

“Games are thus the most ancient and time-honored vehicle for education. They are the original educational technology, the natural one, having received the seal of approval of natural selection”. Written by Chris Crawford, in his book *The Art of Computer Game Design*, this statement proves the importance of games in any aspect of education. Games have been used throughout time as an instrument of instruction for all different aspects of life. Puzzles to learn logic, mathematical games to enhance basic math skills, and even reading games to increase reading ability have all been used successfully to teach children the basic skills that they will need in life. “It logically follows, then, that using computer games is an effective way to teach computing skills, and utilizing course curriculums that teach how to program computer games would invariably teach the basic skills required to program anything” (Baibak and Agrawal, 2007).

Nowadays we are dealing with a reduction in students which are fond of algorithmic and programming. This reduction is reflected not only in the number of students interested learning algorithmics and programming, but also in the quality of the participants in these events. A quite different view is presented when the students have to program a game. They have some inner motivations to consider this game as a challenge making the efforts to find the best winning strategy.

Programming games will endow the students with some skills which will be very useful for their future activities. Nowadays the computer game market is in expansion and the students will be the future programmers and more. As Gordon Novak Jr. (see website) noticed: “Games are good vehicles for research because they are well formalized, small, and self-contained. They are therefore easily programmed. Games can be good models of competitive situations, so principles discovered in game-playing programs may be applicable to practical problems”.

In the first IOI there was a game task, and the game tasks continue to be presented in the IOI tasks sets in a sporadic way. From the first IOI till now there have been 23 reactive and game tasks versus 97 batch ones. There are only two IOIs where two game tasks were presented – IOI’2001, Tampere, Finland and IOI’2006, Merida, Mexico. Perhaps there are two HSC leaders fond of game tasks – Jyrki Nummenmaa, and Cesar Cepeda – who must be followed by others.

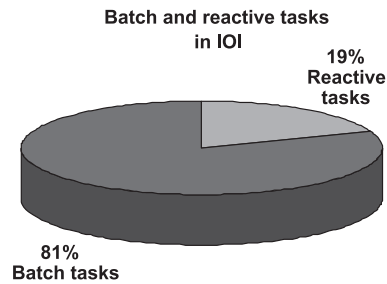


Fig. 1. Batch and reactive tasks in IOI.

According to the IOI 2008 Competition Rules for the reactive tasks the task statements should define among others:

- the interface specification of the “opponent” library,
- explanation of how to interact with the “opponent” library,
- instructions on how to compile their programs with provided “opponent” library.

These are the same characteristics as the game tasks where the player 1 (the contestant) plays against the player 2 (the opponent library).

Programming game tasks are very closely related with research activity. According Susane Epstein (1999): “There are two principal reasons to continue to do research on games ... First, human fascination with game playing is long-standing and pervasive. Anthropologists have catalogued popular games in almost every culture ... Games intrigue us because they address important cognitive functions ... The second reason ... is that some difficult games remain to be won, games that people play very well but computers do not. These games clarify what our current approach lacks. They set challenges for us to meet, and they promise ample rewards.”

Considering the importance of programming games a Games Group has been formed in the University of Alberta which produces high-performance, real-time programs for strategic game-playing (University of Alberta GAMES Group, 2006). The group employs a variety of techniques from many areas of computer science, including artificial intelligence, parallel processing, and algorithm analysis.

At the Stanford University there is a research project by the Stanford Logic Group, part of the Stanford University Computer Science Department. Their AI Magazine article describes the General Game Playing concept and the AAAI GGP competition (AAAI General Game Playing Competition, 2008); a brief GGP Overview is also available. The GGP website contains information about the Logic Group’s research in general game playing, and forms the central resource for General Game Playing Competitions, the first of which was held at AAAI ’05 in Pittsburgh. The website also hosts a GGP Game Manager, allowing General Game Players to connect and play single or multi-player games online, in order to help them to prepare for future competitions.

5. Conclusions

The reactive tasks and especially game tasks at the IOI must be considered as very useful tool for making this event more attractive to the students. These kinds of tasks are very challenging, and students are very motivated to undertake their programming. These kinds of tasks are very close to real life situations, and the students do not spend too many efforts understanding or remembering them. The game tasks are in harmony with the story describing them and do not need too much input data. More attention must be paid to this kind of tasks at the IOI.

References

- AAAI General Game Playing Competition (2008).
<http://games.stanford.edu/competition/competition.html>
- Baibak, T. and Agrawal, R. (2007). *Programming Games to Learn Algorithms*. American Society for Engineering Education.
- Ginsberg, M.L. (1998). Computers, games and the real world. *Scientific American: Exploring Intelligence* (special issue – Winter 1998). <http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/1198ginsberg.html>
- Gordon Novak Jr. Web-site. (n.d.)
<http://www.cs.utexas.edu/~novak/cs34312.html>
- Epstein, S.L. (1999). Game playing: The next moves. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. Orlando, FL.
- University of Alberta GAMES Group (2006).
<http://www.cs.ualberta.ca/~games/>



I. Ninka is full professor in data structures and algorithms, artificial intelligence at Tirana University, Albania. He is head of Computer Science Department, a member of the Albanian National Commission for Olympiads in Informatics. Since 1993 and was head of the Commission (1999–2001). He was member of the organizing team of BOI (2001), president of the SC of Balkan OI'2001, leader of Albanian national team for IOI'1999–2008. He is author of more than 20 scientific papers, 6 university textbooks and 12 textbooks for secondary schools.